Secure Surveillance using SMIL

Naren Kodali[1], Duminda Wijesekera[1] and J. Bret Michael[2]
[1]Center for Secure Information Systems,
George Mason University, Fairfax, VA 22030-4444.
[2]Department of Computer Science,
Naval Postgraduate School, Mail Code: CS/Mj Monterey, CA 93943.
{nkodali, dwijesek}@gmu.edu, bmichael@nps.navy.mil

Abstract

The role of multimedia and user interactivity has increased in recent years. User interaction is an important component of emerging multimedia systems, and the methods of interaction will become increasingly complex as they are being used in more diverse applications. One such application is surveillance, both in the civil as well as the military domain. The real-time three-dimensional stereovision adds sense of immersion and makes imagery closer to those produced by our own eyes as described in [1]. In our paper we propose a framework to capture an unfolding scenario as it happens and transmit the captured media using SMIL securely to a group of users with varied privileges. Apart form enforcing privileged access our model also ensures integrity of the content in transit through encryption

1. Introduction

Consider the following situation wherein a court of law is examining a critical piece of evidence, where an audio-visual feed from the surveillance camera in a parking lot. This camera feed shows an accused firing at a person. In reality the shooting could have been a defensive action to avoid the victim trying to stab the shooter from behind. A camera form a single stationary angle is insufficient to capture this scene in its complete detail. Having multiple cameras simultaneously recording this scene from different angles would have revealed the total scene as it unfolded. These multiple feeds could be instantly combined and processed into a real-time interactive three-dimensional video. The interactivity and the ability to switch view points would have given the investigators and the jury the liberty to look around the surroundings of scenario as seen from multiple points of view before making their conclusions.

--- ----------------

Partly supported by NSF under grant CCR-0113515, Center for Secure Information Systems at GMU and Prof. S. Jajodia

In addition to the capturing the situations as they unfold, there is a need to effectively transmit the captured detail to intended recipients in a secure way. Access control and Integrity are the key factors because we do not want any unauthorized people to access the content as well as we want it to be safe from tampering and unwanted modification. Synchronized Multimedia Integration Language (SMIL)[4] is a XML like language that enables us to integrate real-time media and enforce a security policy that is contained within the SMIL document.

In this paper we propose a methodology for implementing access control by embedding the authorization rules in the SMIL document created for transmitting media that is captured during surveillance. The primary motive is the actual transmission and display of mayday information (via surveillance) using low-cost off-the-shelf software (Web browser and scripting) and open standards (SMIL, XSL, and HTTP).

Providing such features require many technical challenges to be solved and integrated. Various trade-offs and compromises will be required so that different pieces of existing technology can be artfully combined to create a new and advanced multimedia environment. These tradeoffs need to be evaluated in light of *quality to effectiveness tradeoffs* of enhanced QoS parameters [2] associated with looking around and stereoscopy. Since proposed services support numerous audio and video streams, substantial bandwidth will be required for memory, I/O and the network so that the key functionalities like mediate stream acquisition, processing, transmission and display site management utilize system resources efficiently, but satisfy the user's qualitative needs.

The remaining paper is organized as follows; Section 2 discusses the related work. Section 3 describes surveillance, Section 4 introduces SMIL, Section 5 describes Multiparticipant Stereo video, Sections 6 and 7 explain the access control and encryption, Section 8 summarizes the future work and Section 9 concludes the paper.

2. Related Work

Schmidt [1] has defined an architecture for distributed multi-participant and interactive multimedia with a structure that enables multiple users to share media streams within a network distributed environment. Multimedia streams, consisting of audio and video originating from various sources can be combined to provide media clips that accommodate look-around capabilities. We use this model for the purpose of surveillance and the relevant architecture is described in the later sections.

Regulating access to XML documents of textual nature has been actively researched in the past few years after it has been accepted that XML is soon going to become the de-facto standard for the World Wide Web. Bertino et al [6,7,8,9], Damiani et al [13,14,15,16] and Gabillon et al [5] have suggested access control models for controlling access to XML documents.

Bertino et al [7,8,9], have developed Author-X, a Java based system to secure XML documents that supports access control policies at various granularities and user credentials. Author-X encodes security policies for a set of XML documents in an XML file referred to as the policy base. They permit both permissions and prohibitions. This feature enables the user to specify exceptions with ease as opposed to creating a set of XML documents and document type definitions (DTD's). There are conflict-resolution and default strategies to address over specification and under specification respectively.

Damiani et al [13,14] developed an access control model where the tree structure of XML documents are exploited using XPATH [17] to control access at different levels of granularity. Here, the smallest protection granularity is a node, and permissions or prohibits to a node can be defined. Fine-grained accesses are specified using XML markup where a subject is a user who is generally a member of one or more user groups, and an object is any node on a XPATH tree.

Gabillon et al [35] have suggested an alternative to Damiani et al [13,14], where authorization rules related to a specific XML document are first defined on a separate authorization sheet (style sheet). This sheet is then translated to an (eXtensible Stylesheet Language) XSL sheet. If a user requests access to the XML document then the (XSL Transforms) XSLT [18] processor uses the XSLT sheet to compute the view of the XML document with appropriate rights.

Damiani et al [15] have proposed a model for selectively controlling access to images. The XML based data model of SVG is exploited to control access to graphic information on the web. The primary focus is the vector image data, which has emerged as an alternative to contemporary raster image formats.
But none of these proposals are completely satisfactory for multimedia. They primarily address textual documents and exploit the granular structure of XML documents. Although [15] addresses feature protection of XML format based images, its primary focus is controlled dissemination of sensitive data within an image. Multimedia for various reasons as discussed above has to be treated differently in our framework we use an XSLT processor to deal on our authorization mechanism as well as propose a variation to the encryption model of Bertino et al [8,9].

Boneh et al [10] proposed a public key encryption scheme for multimedia in which there is one encryption key and multiple decryption keys. Multimedia content is encrypted and distributed over a broadcast channel to subscribers, and the decoder box has the secret decryption key, which decrypts the public broadcast. Problems of users sharing decryption keys or groups of users colluding to create a new decryption key are addressed and a traitor-tracing scheme is implemented to identify traitors. Fiat et al [11] proposed a dynamic traitor-tracing scheme, where systems with conditional accesses are safeguarded using watermarking. The root of any potential piracy is traced and it is disconnected from the regular transmission without harming legitimate users. Using their approach it is also possible to gather forensic evidence to incriminate the abusers.

Both [10] and [11] address the issue of multimedia and are very effective for commercial Cable TV applications, but cannot be adapted "as is" in the context of multimedia presentation over the any private network or the internet because of the watermarking and the variations of the media clips generated thus are huge and would be very server intensive, thereby could result in a bottleneck.

3. Physical Surveillance and Emergency Response.

Current surveillance systems, such as those available on common desktops and the Internet consists of audio and video either collected live, such as a camera mounted on a tall building, or served out of a storage device with or without an accompanying audio stream. But in order to implement surveillance on a commercial scale for real-time applications, rates as minimal as 30 video frames per second and synchronized audio and video delivery are the need of the hour.

Let us consider the scenario in which we use multiple cameras and microphones to capture an accident at a frequent accident zone in Fairfax, VA. The city council has decided to declare Vinerva Circle as a Level II danger zone, because of the high frequency of accidents that have been recorded in the past year. The council has decided to install multiple cameras and microphones that constantly provide real time audio and video for physical surveillance. An Emergency Help Center is created wherein the feeds from the camera and audio

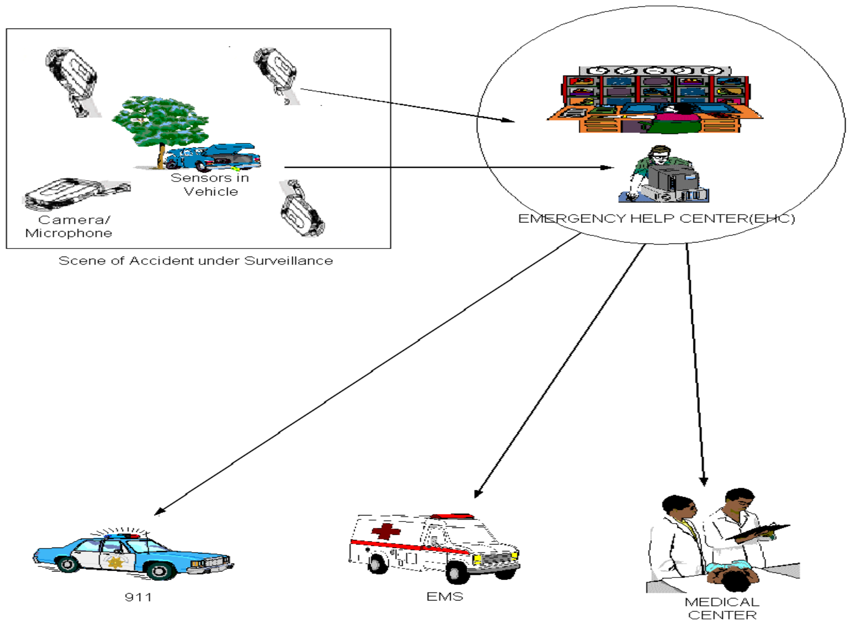

Figure 1: An Example Scenario

devices are received and processed. Let us assume that every vehicle-owning citizen of the city is a mandatory subscriber. Each individual subscriber is provided with a ***sensor*** that has to be installed on the vehicle. This ***sensor*** is used to detect and record vehicle status before, during, and after the collision, e.g. change in velocity, angle of impact, airbag deployment, rollover, and ending position. At service initiation users are required to provide critical as well as some personal information (age, domicile, blood type etc) upon which appropriate decisions can be made in the event of an emergency. This and any other relevant information obtained by filling a form are the grouped together as a database. The EHC then processes some of this information to uniquely distinguish between subscribers based on their rights, requests and privileges into a list called the *access control list* (the functionality of which will be described in later sections).

When an accident occurs there are three distinct types of information that can be obtained from the equipment installed for surveillance as shown in Figure 1:

1) The information obtained from the sensors installed in the subscriber's car
2) The live video feed from multiple cameras installed at strategic angles.
3) The real-time audio associated with its camera at the place of the accident.

Information obtained form the sensors are used for Emergency Response and the feeds from the camera and the microphone are used to analyze the situation later. The medical data must have been provided in advance by the EHC subscriber, with the provision that it will be released to appropriate medical personnel only in emergency situations. In the future, real-time traffic information could be tied into this system to take current congestion and special road conditions into account.

All this information obtained is sent to the EHC through secure and dedicated channels as soon as an accident occurs. Then EHC will then add vehicle information and medical data about the probable or confirmed occupants of the vehicle from its database. The subscriber must previously have authorized release of the medical data in case of emergency.

The EHC could send information to three different places:

1) 911 Control Room
2) EMS Dispatcher
3) Personal Medical Control Doctor/Hospital

Firstly, the enhanced data is transmitted to the 911-control room.

In our model we create a voice link to a car occupant has been established and the call-taker learns additional information (e.g. there is a child in the back seat). The EHC will then send information to the EMS dispatcher all the data it has obtained (with relevant medical information), plus any new or corrected information obtained by voice link and typed into the data system. In a real system, the EHC would have the choice of many possible recipients of the information in which case he might have to either broadcast or multicast the information securely.

The EMS dispatcher views relevant portions of the data integrated with a Computer-Aided Dispatch (CAD) system, dispatches appropriate response vehicles and send s back information to the EHC in case of change/addition of information. The EHC then transmits the media file to medical control, again with possible additions or corrections Medical control receives all the collision and medical data, including additions from the EHC and the EMS-dispatcher.

.

In the model all data and media items that arrive at the EHC are stored in the form of a Synchronized Multimedia Integration Language (SMIL) file. SMIL start and end tags identify each piece of information and block of information. The same SMIL file is passed from each location to the next in the sequence, while adhering to rights and privileges of the subscribers by enforcing access control as will be discussed in the later sections. All transmission of data files between intermediate stations in the network use standard hypertext transfer protocol (HTTP) web protocol. When each computer in the process receives a data file, it opens a standard web browser window, and then displays the XML file in the browser window using a customized extensible Stylesheet Language (XSL) style sheet.

The style sheet selects which data items to display and the format for displaying them. Data elements in the file may be made available on a need-to-know basis to protect the privacy of subscriber information such as the time the incident occurred, the EHC that relayed the call, the location of the incident (looked up by a map data base based on the information from the sensors from the given GPS-supplied latitude and longitude), some information about the collision, and the names of the probable occupants shows up on the computer's browser window after receiving the XML data file and displaying selected portions of it with a style sheet.

At the EMS dispatcher's screen, the updated data file in SMIL format has been received, but a different XSL style sheet has selected different portions of the data to display. For example, the display may include more detailed information about the location of the car and its make, model, and license. The number of occupants field reflects the change made by the call-taker at EHC.

Although the same SMIL data file is transmitted, many different style sheets are used to extract privileged and protected information by following an access control model. For e.g.: medical information present in the file but not displayed to the at all intermediate stations.

While information is sent to the medical unit, the private medical information about the occupants such blood type, allergies, current medications, and current health are listed, with unusual items shown in red. This information could be useful to medical and paramedical personnel. This medical history information and contact information for each occupant's doctor and additional information regarding medical history file are maintained in the EHC database but are not included in the mayday data transmission. Instead, they may be retrieved by an authorized medical provider who has been authorized by the victim after the SMIL document has been transmitted.

4. SMIL

SMIL [1] is an XML-like language developed by W3C to author multimedia presentations. The major components of multimedia such as audio, video, text and images can be integrated and synchronized to form a presentation or a media clip. The distinguishing features that separate SMIL from XML are well-defined syntactic constructs for timing and synchronization of media streams that allow fine-grained synchronization. SMIL also has syntax for spatial layout including constructs for non-textual and non-image media and hyperlink support for time-based media making SMIL adaptable to varying user and system characteristics.

The important integration features offered by SMIL are the synchronization constructs such as <seq>, <excl> <repeat> and <par>. They are used to hierarchically specify synchronized multimedia documents. The <seq> element plays the child elements one after another in sequential order. In such a hierarchical specification <excl> element plays one child at a time, but does not impose any order. The <par> element plays child elements as a group (allowing *parallel* playback). In SMIL, every element has a beginning (*begin*) and a *simple duration*. *Begin* can be specified in various ways, an element can begin at a given time, or based upon when another element begins or when some event occurs. *Simple duration* defines the basic presentation time of an element. Elements can be specified to repeat their simple duration a fixed finite number of times. The simple duration and any repetitions can be combined to determine the so-called *active duration*. An element *becomes active* when it begins its active duration, and *becomes inactive* when it ends its active duration. When an element's active duration terminates, the element can either be removed from the presentation or *frozen* meaning it is held in its terminal state: e.g., to fill dead-times in the presentation. Attributes that control these synchronization specifications can be applied not only to media elements, but to complex constructs (par, seq etc) as well.

```
<smil>
  <head>
    <layout type="text/smil-basic">
    <channel id="video1" left="20" top="50" z-index="1"/>
    <channel id="text1" left="20" top="120" z-index="1"/>
    <channel id="video2" left="150" top="50" z-index="1"/>
    <channel id="text2" left="40" top="70" z-index="1"/>
    <channel id="video3" left="60" top="120" z-index="1"/>
    <channel id="text3" left="70" top="110" z-index="1"/>
    </layout>
  </head>
 <body>
<seq>
<sensor>
<text src = "Input from EHC" channel = "text1"/>
<text src="Input from EHC" channel="text2"/>
 </sensor>
<seq>
        <img   src="Accident"/>
      <par>
        <camera1>
        <video src="CameraA" channel="video1"dur="45s"/>
        <audio src="Camera A audio"/>
        </camera1>
     </par>
     <par>
        <camera2>
        <video src="Camera2 " channel="video2"/></a>
        <audio src="Camera2 audio"/>
        </camera2>
     </par>
     <par>
        <camera3>
        <video src="Camera3 video" channel="video3"/>
        <audio src="Camera3 audio"/>
        </camera3>
     </par>
 </seq>
 </body>
</smil>
```

Figure 2: SMIL representation of information received at EHC

This allows, for example, an entire sequence to be repeated, and to be coordinated as a unit with other media and complex presentations such as media clips. Audio and video, which constitute basic multimedia stream types, can be combined to form a media clip by using the synchronization constraints to specify the exact synchronization we need. SMIL has been constantly developing and becoming rich in features and is supported by many present–day commercial browsers and was developed to work with many commercial and free media players.

An example SMIL document given in Figure2 shows how the information received at the EHC is combined to make a multimedia presentation. As the figure shows there are four forms of media video, audio, text and image that are integrated and synchronized using <seq> and <par> and duration. The primitive media entity is a frame, many of which are grouped together to form scenes. The audio, video and text frames (if any) run in parallel to combine and render a scene for a clip. A group of scenes constitute a shot, the EHC defines how many and which scenes constitute a shot. In our model we chose a "shot" as the level of granularity on which we impose access control and encryption. As shown in the figure above, the tree structure of SMIL is depicted for a media clip at the shot, scene and frame level.

5. Multi-participant Stereo Video

After the emergency response phase, we would now process the real-time video and audio received form the scene as it unfolded. The EHC will have control over the cameras installed at the location and would be able to switch from one to another real-time. Further the cameras can be rotated and made to "*look-around*" thereby providing a 3-d view and sense of immersion. We would have multiple feeds from different cameras and the EHC needs to clearly edit the irrelevant information and make an informative media clip out of it. All the feeds are combined to form a three-dimensional video of the scenario with look-around capability. We use SMIL and its timing constructs as discussed in Section 5 to make a multimedia clip form the information at hand and can transmit it securely to any authorized station with the access privileges appropriately enforced. The Figure3 below shows a SMIL structure of a media clip that is generated based on the information received and is carefully divided into shots. Each shot is based on a set of authorization rules that are derived from the *access control list.*

```
<smil>
<body>
 <seq>
        <img   src="Media Clip : Accident No: 6345"/>
      <par>
        <shot1>
        <video src="Camera1" dur="45s"/>
        <video src="Camera3" dur="23s"/>
        <video src="Camera2" dur="4s"/>
        <audio src="Camera1  "/>
        <text src = " sensor" />
        </shot1>
     </par>
     <par>
        <shot2>
        <video src="Camera3" channel="video2"/></a>
        <audio src="Camera3 audio"/>
        <text src="sensor" dur "8s " channel="text2"/>
        </shot2>
    </par>
              xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
              xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
              xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

 (Similarly various shots with pertinent combinations as decided by
EHC are made to form the 3-d media clip>

</body>
</smil>
```

Figure 3: Media Clip generated as a combination of "shots".

6. Access Control Model

After the process of creating the shots and the SMIL document we need to specify authorization control rules made based *access control list*, which was generated based on the subscriber information and their rights and preferences. The authorization rules are formulated and represented on XML style sheet called XPAS (XML Policy Authorization Sheet). This is a typical XSL sheet that is in accordance to [18].

The time-container tags encompass the frames and scenes and the level of granularity is the shot, which means that we would divide the entire document into a sequence of individual shots upon which conditional and privileged access and encryption can be implemented to enforce security.

Any SMIL document can be represented as a tree with "n" number of nodes. XPATH [15] is a regular language for denoting nodes of an XML document. Since the syntactic structure of SMIL is similar to XML syntax we use XPATH to address individual nodes. The authorization rules specify access privileges granted to the user when he subscribes to the service. In these rules, *subjects* are subscribers of the EHC and he/she is uniquely identified by some attributes as discussed in earlier sections An *object* is any node in a XPATH tree. The most helpful feature of XPATH it facilitates pattern matching using regular expressions. This is helpful in identifying objects that meet and adhere to the specified policy.

A pattern search is run on the XPATH tree to yield what shots he/she can possibly see. An XSL Transform [XSLT] [18] is used to convey the authorization rules to the SMIL document and extract the nodes that obey the authorization rules. As mentioned earlier each intermediate station that the information passes through has a different XSLT pertinent to the authorization and access privileges that it enjoys The transformation is the actual interface that relates the authorization stylesheet to the SMIL specification of the media clip. XSLT understands the rules as patterns and using XPATH parses through the SMIL document and retrieves the appropriate shots.

In our example we consider three types of classifications based on the receiving station. These classifications are based on the "need-to-know" of each station and the personal preference and rights of the subscriber. All pertinent controls that are decided based on the access policies and security considerations can be very precisely represented using SMIL and thereby essentially capture even minute detail that is necessary. A list of all permissible shots can de derived and represented in the SMIL specification. The SMIL document in Figure4 below shows how we specify the access constraints in SMIL.

7. Encryption Model

The encryption model is analogous to the real world methodology [10] adopted when broadcasting secure data (e.g. bank information, stock quotes) via the Internet. A *smartcard*, which contains the authorization rules based on the *access control list*, is installed on the all the intermediate stations. It could be a hardware device or a software program to achieve the same result.

```
<smil>
<accesspolicy>
<station1 = "EMS">
  <shot1>
  <shot2>
  <shot7>
</station1>
<station2 = "Medical Control">
  <shot2>
  <shot3>
  <shot4>
</station2>
(We can add other intermediate stations here)
</accesspolicy>
<body>
 <seq>
        <img   src="Media Clip:Accident No: 6345"/>
      <par>
        <shot1>
        <video src="Camera1" dur="45s"/>
        <video src="Camera3" dur="23s"/>
        <video src="Camera2" dur="4s"/>
        <audio src="Camera1  "/>
        <text src = " sensor" />
        </shot1>
     </par>
     <par>
        <shot2>
        <video src="Camera3" channel="video2"/></a>
        <audio src="Camera3 audio"/>
        <text src="sensor" dur "8s " channel="text2"/>
        </shot2>
    </par>
             xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
             xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
             xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
</body>
</smil>
```

Figure 4: The media clip with embedded Access policy.

It has an inbuilt Cryptix Parser that is programmed in firmware (or in software) to handle the decryption process that is described below. Decrypting the associated shots in a broadcasted environment is the primary functionality of the intelligent Cryptix parser within the *smartcard*.

The encryption model uses both symmetric encryption and Public Key encryption for two mutually diverse events to serve varying purposes.

Step1:

The SMIL format of the media clip is divided into ***shots*** as discussed earlier, and since our element of granularity is a "shot"; we encrypt each shot with a unique Symmetric Key. In the SMIL document shot1 is encrypted with **SymmetricKey**$_{shot1}$ and shot2 with **SymmetricKey**$_{Shot2}$ and the process is repeated for all the shots within the document. All the encrypted shots have a corresponding symmetric decryption key (which is the same as the encryption key). Based on the subscriber authorization rules a finite number of symmetric decryption keys strictly corresponding to the allowed shots are grouped together as a vector of keys (***keyvector)***. This distinct keyvector for a particular station would be able to release all or a subset of shots depending on the their nature and the authorization rules decided by the authoritative body for this intermediate station.

Step 2:

When information is sent to an intermediate station a keyvector is generated and is then encrypted with the **PublicKey** of the particular station. The format of the keyvector is as follows:

PublicKey of Station ((SymmetricDecryptionKeyNumber (Shot Number)…....
…SymmetricDecryptionKeyN (ShotN)))

We do not encrypt the entire media clip with a Public Key, because all the shots are encrypted, any unintended or malicious recipient cannot view the media clip even though they have access to it, thereby eliminating the need for such a methodology.

The keyvector (a finite set of valid symmetric decryption keys) encrypted with the PublicKey of the recipient station is unicast to the intermediate station prior to the actual broadcast of the media clip. Eventually when the media clip arrives, the decryption process is initiated and completed by the CryptixParser.

```
<smil>
<accesspolicy>
<station1 = "EMS">
  <shot1>
  <shot2>
  <shot7>
</station1>
        xxxxx
        xxxxx
(We can add other intermediate stations here)
</accesspolicy>
<body>
 <seq>
         <img   src="Media Clip: Accident No: 6345"/>
       <par>
<Shot1 is encrypted with SymmetricKeyShot1 >
         <shot1>
         <video src="Camera1" dur="45s"/>
         <video src="Camera3" dur="23s"/>
         <video src="Camera2" dur="4s"/>
         <audio src="Camera1  "/>
         <text src = " sensor" />
         </shot1>
      </par>
      <par>
<Shot2 is encrypted with SymmetricKeyShot2 >
         <shot2>
         <video src="Camera3" channel="video2"/></a>
         <audio src="Camera3 audio"/>
         <text src="sensor" dur "8s " channel="text2"/>
         </shot2>
     </par>
Shot3 and all other shots upto ShotN are encrypted with the respective
SymmetricKeyShot (1-N)
                xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
</body>
</smil>
```

Figure 5: Media Clip with the shots individually encrypted.

There are unique decryption key vectors specific to each individual transmission, that encompass the diverse privileges of both the victim/subscriber and the remote station so we transfer the load from the server onto remote intermediate station

The server at the EHC takes the following steps:

1. Create the *keyvector*, encrypt with the PublicKey of the client (station) and deliver (unicast) it to the *smartcard* on the client machine.
2. Broadcast the encrypted media clip.

Each client (station) receiving a key vector takes the following steps:

1. Decrypt the received *keyvector* and install on the *smartcard*.
2. Decrypt the encrypted parts of the media clip with available symmetric decryption keys.

In a broadcast environment where there is more than one user, which is generally the case, there is a significant delay between shots in the media clip. This is because the encrypted shots arrive sequentially and there is a delay associated with the decryption process and the sequential order of the viewable shots. Real time system issues such as the ability of the client to download the keys as well as the encrypted media clip well in advance, so as to decrypt and **buffer** the valid shots in a sequential order to minimize any significant delay is an open issue that we intend to address in very soon.

8. Future Research

The model we discussed above can be enhanced to encompass enhanced user interactions such as fast forward, rewind, fast play: controls found on home VCR's. This enhancement would enable us to carefully examine the implicit detail and allow us to make assertions based on the media information that we capture. This is precisely the focus of our future research.

9. Conclusion

We have provided a model for advancing state of the art in audio-video services to provide 3-dimensional stereoscopic multimedia, where the source has a collection of cameras and microphones. The information (both text and video) is securely transmitted as SMIL documents with access control and encryption. We have provided a wider view with multiple perspectives. Secondly, our provides much more depth information than the view of a single camera and the cost of providing such a secure service is affordable and can be deployed with sufficient infrastructure.

References

[1] B. K. Schmidt "An Architecture for Distributed, Interactive, Multi-Stream, Multi-Participant Audio and Video". Technical Report No CSL-TR-99-781, Stanford Computer Science Department.

[2] D. Wijesekera and J.Srivastava, "Quality of Service Metrics for Multimedia" in Multimedia Tools and Applications, Vol 2, No3 1996, pp. 127-166.

[3] A.G. Stoica and C. Farkas. "Secure XML Views" in *Proc. IFIP WG11.3 Working Conference on Database Security*, King's College, Cambridge, England.

[4] J. Ayers et al. "Synchronized Multimedia Integration Language (SMIL 2.0)". World Wide Web Consortium (W3C). http://www.w3.org/TR/smil20/ (August 2001).

[5] A. Gabillon, E. Bruno. Regulating Access to XML documents. in *Proc. IFIP WG11.3 Working Conference on Database Security*, Niagara on the Lake, Ontario, Canada, July 15-18, 2001

[6] B. Carminati, E. Bertino, E. Ferrari. "XML Security" in Information Security Technical Report, Vol 6,No 2(2001) Pages 44-58..

[7] E. Bertino, E. Ferrari S. Casatano "Securing XML Documents with Author-X" in IEEE Internet Computing, vol 5,no3 May/June 2001

[8] E. Bertino, M. Braun, S. Castano, E. Ferrari, M. Mesiti. "AuthorX: A Java-Based System for XML Data Protection". In Proc. of the 14th Annual IFIP WG 11.3 Working Conference on Database Security, Schoorl, The Netherlands, August 2000

[9] E. Bertino, S. Castano, E. Ferrari and M. Mesiti. "Specifying and Enforcing Access Control Policies for XML Document Sources". World Wide Web Journal, vol. 3, n. 3, Baltzer Science Publishers.

[10] D. Boneh, M. Franklin "An Efficient Public Key Traitor Tracing Scheme" in Eurocrypt 99.

[11] A. Fiat, T. Tassa "Dynamic Traitor Tracing" Pages 211-223The Journal of Cryptography, April 2001.

[12] D.Wijesekera, N.B.Kodali, S.Jajodia "Regulating Access to SMIL formatted Pay-per-view Movies" to appear in CCS, Workshop on XML Security, 2002.

[13] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, P. Samarati, "Securing XML Documents," in *Proc. of the 2000 International Conference on Extending Database Technology (EDBT2000),* Konstanz, Germany, March 27-31, 2000.

[14] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, P. Samarati, "Controlling Access to XML Documents," in IEEE Internet Computing, vol. 5, n. 6, November/December 2001, pp. 18-28.

[15] E. Damiani, S. De Capitani di Vimercati, E. Fernandez-Medina, P. Samarati "An Access Control System for SVG Documents" in *Proc. IFIP WG11.3 Working Conference on Database Security*, King's College, and Cambridge, England

[16] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, P. Samarati "XML Access Control Systems: A Component-Based Approach" in *Proc. IFIP WG11.3 Working Conference on Database Security*, Schoorl, The Netherlands, August 21-23, 2000.

[17] J. Clark et al.. "XML Path Language (XPath) Version 1.0". World Wide Web Consortium (W3C). http://www.w3c.org/TR/xpath (November 1999).

[18] J. Clark et al. "XSL Transformations (XSLT) Version 1.0". World Wide Web Consortium (W3C). http://www.w3c.org/TR/xslt (November 1999).

Biographies:

Naren B.Kodali is currently pursuing PhD degree at George Mason University. His area of research is Multimedia security and QoS of continuous media. LLC. He received his MS (Computer Science) form George Mason University in 1999.Earlier he has worked in various positions at Ecutel Inc and Navtech, LLC.

Duminda Wijesekera is an Assistant Professor of Information Systems at George Mason University. His primary focus of research is information and data security, continuous media security and constructive modal logic. Prior to joining GMU, he was at Honeywell Space Systems and University of Wisconsin. He received two PhD degrees in Mathematics (Cornell '91) and in Computer Science (University of Minnesota '98).

J. Bret Michael has been an Associate Professor of Computer Science at the Naval Postgraduate School since 1998. His most recent research on the subject of distributed computing has been focused on issues pertaining to information security, system architecture, and software system safety. Prior to joining NPS, he conducted research at the University of California at Berkeley on the technical feasibility of fully automating the operation of dual-mode passenger and commercial vehicles on limited-access highways. He received his Ph.D. degree from George Mason University's School of Information Technology and Engineering in 1993